

**Use of the Universal Serial Bus as an Internal Architecture within IDE Disk
Array**

By
Thomas Bolt
and
Bill Moon

Field of the Invention

The present invention related to data storage devices, and in particular to
disk arrays.

Background of the Invention

The Integrated Drive Electronics (IDE) bus is commonly implemented as a
secondary bus to interface mass storage devices such as hard disk drives, floppy
disk drives, and CD ROM drives. The IDE bus remains the most widely-adopted
bus architecture for mass storage devices in personal computer systems. An
IDE controller can support a maximum of up to two IDE devices. If two IDE
devices connect to one IDE controller, one device is designated as the "master"
and the other as the "slave," according to the IDE protocol. Also, if two IDE
controllers are incorporated simultaneously into the same computer, one bus
controller is designated as the "primary" with the other as the "secondary." The
master/slave and primary/secondary designations facilitate the complex
negotiations between multiple IDE devices and buses.

Existing disk drive arrays utilize multiple IDE disk drives for cost savings
compared to SCSI or Fibre Channel disk drives. When building storage arrays
using IDE disk drives, several limitations of the IDE bus become immediately
apparent. The IDE bus only allows two IDE devices per bus, requiring the arrays
to include many IDE buses. The IDE bus is limited to a length of 18 inches for
high speed applications (UDMA 66, UDMA 100, UDMA 133). The IDE bus has

1 no provisions for "hot plugging" of devices. And, the IDE bus is a parallel bus so
2 that trace routing becomes complex.

3
4 Such conventional storage arrays use multiple IDE controllers, ASICs, or
5 FPGAs. Owing to the bus length limitations of IDE, such storage arrays must
6 locate the controllers as close as possible to the disk drives themselves, typically
7 on a center plane of the storage enclosure. This leads to a highly complex and
8 active center plane, severely impacting the field serviceability of the devices.
9 Furthermore, additional circuitry must be added to handle "hot plugging" of failed
10 and replacement disk drives because "hot plugging" is outside the scope of IDE
11 bus specification, subject to incompatibilities with IDE disk drives.

12 13 **Brief Summary of the Invention**

14 The present invention alleviates the above limitations. One type of bus
15 architecture, the Universal Serial Bus (USB), typically supports peripheral
16 devices external to a computer, such as video cameras, modems, joysticks, a
17 keyboard and mouse, and other input/output devices via a special USB
18 connector. In addition to simultaneously supporting a large number of peripheral
19 devices, the USB architecture also supports hot plugging of peripheral devices.
20 A Universal Serial Bus operates using a central USB controller to arbiter and
21 route all bus traffic.

22
23 The USB bus has relatively low throughput compared to Fibre Channel or
24 ultra SCSI, however due to the low cost of USB components, it is highly desirable
25 in applications where either a large number of relatively slow devices are used,
26 or, a large number of relatively fast devices are used but with only a subset
27 active at any given time.

28
29 In one embodiment, the present invention provides a system including one
30 or more IDE devices, each IDE device connected to a USB to IDE controller,
31 wherein the USB to IDE controllers are connected to one or more USB

1 controllers. USB is a high speed low cost serial bus architecture, that provides
2 "hot plugging", supports up to 127 devices, and supports cable lengths up to 15
3 meters. The USB architecture is a point to point, shared bandwidth architecture.
4 Hub devices may be used to connect additional devices to a USB controller.
5

6 In one version, a data storage system according to the present invention
7 includes multiple IDE disk drives, wherein each disk drive is connected to a USB
8 to IDE controller. Within the data storage system (array), USB to IDE controllers
9 are integrated into disk drive carriers. The logical interface presented to the
10 storage array is strictly USB, and the insertion or removal of the drive carrier
11 corresponds directly to USB device insertion/removal from the USB bus. Hubs
12 can be used to associate up to 127 USB devices to a single USB controller, and
13 multiple USB controllers may be utilized to increase overall system. Using the
14 simple four wire interface of USB and the extend lengths of USB relative to IDE,
15 a passive center plane design for the storage array is made possible by the
16 present invention. Using USB as an internal interconnect allows the use of
17 inexpensive IDE disk drives within a storage array with simplified cabling/signal
18 routing, "hot plugging", and passive center plane.
19

20 **Brief Description of the Drawings**

21 These and other features, aspects and advantages of the present
22 invention will become understood with reference to the following description,
23 appended claims and accompanying figures where:

24 FIG. 1A shows an example block diagram of an embodiment of a system
25 according to the present invention;

26 FIG. 1B shows an example block diagram of another embodiment of a
27 system according to the present invention;

28 FIG. 2 shows a perspective view of a schematic diagram of an example
29 data storage system according to another aspect of the present invention;

30 FIG. 3 shows a more detailed view of the disk drives in FIG. 2;

1 FIG. 4 shows a block diagram of the architecture of the data storage
2 system of FIG. 2; and

3 FIG. 5-6 show flowcharts of the steps of different embodiments of the
4 method of using the Universal Serial Bus as an Internal Architecture within a
5 storage system.

6
7 **Detailed Description of the Invention**

8 The present invention utilizes the Universal Serial Bus (USB) serial bus
9 architecture as described herein. The USB architecture, typically supports
10 peripheral devices, such as video cameras, modems, joysticks, a keyboard and
11 mouse, and other input/output devices via USB connectors. In addition to
12 simultaneously supporting a large number of peripheral devices, the USB
13 architecture also supports "hot plugging" of peripheral devices. A Universal
14 Serial Bus operates using a central USB controller to arbiter and route all bus
15 traffic.

16
17 For example, USB 2.0 is a high speed (480 megabits/second) low cost
18 serial bus architecture, that provides "hot plugging", supports up to 127 devices,
19 and supports cable lengths up to 15 meters. Further, the USB architecture is a
20 point to point, shared bandwidth architecture, wherein USB hub devices may be
21 used to connect additional USB devices to a USB controller.

22
23 Referring to FIGs. 1A-B, example block diagrams of two embodiments of a
24 USB system 10 according to the present invention is shown. The system 10
25 comprises one or more IDE devices 12 (as used herein, IDE device comprises
26 an IDE/ATA device), one or more USB-to-IDE controllers 14 (USB-to-ATA bridge
27 or protocol converter) corresponding to the IDE devices 12, optionally one or
28 more USB hubs 16 as needed, and one or more USB controllers 18 (preferably
29 USB 2.0), that can be connected to a processor (CPU) 20, interconnected as
30 shown. A USB bus 11 is the connection between the USB controllers 18 and
31 the USB HUBS 16, or the connection between the USB hub 16 and the USB-to-

1 ATA bridges 14, or a direct connection between the USB controller 18 and the
2 USB-to-ATA bridge 14. The connection 15 between the USB controller 18 and
3 the CPU 20 can be a PCI bus.

4
5 Each IDE device 12 can comprise any of several devices such as disk
6 drive (HDD), CD ROM, CR RW, DVD, Tape device, etc. Therefore, the present
7 invention is not limited to a disk drive (HDD) utilized as an IDE device in the
8 description herein. Further, different IDE devices can be used in the system.

9
10 Referring to FIG. 2, the present invention is shown in one embodiment as
11 a data storage array 24, comprising multiple IDE disk drives 12, each connected
12 to an USB to IDE controller 14 (USB-to-IDE bridge or protocol converter). The
13 USB to IDE converters 14 can be connected to the USB controllers 18 directly, or
14 via a hub 16, as shown in FIGs. 1A-B. In the example of FIG. 2, within the
15 storage array 24, the USB to IDE controllers 14 are integrated into disk drive
16 carriers 22 (e.g., FIG. 1B). Each IDE disk drive 12 is connected to a USB
17 controller 18 or USB hub 16 via an USB to IDE controller 14, thereby appearing
18 as a USB device to the USB controller 18. The logical interface presented to the
19 storage array 24 is USB, and the insertion or removal of a disk drive carrier 22
20 corresponds directly to USB device insertion/removal from the USB bus ("hot
21 plugging").

22
23 Further, multiple USB controllers 18 may be utilized to increase overall
24 system bandwidth. Using the simple four wire interface of the USB architecture
25 and the extend lengths of USB connections relative to IDE, the present invention
26 allows design of a passive center plane design for the storage system 24.

27
28 In one version, no cables are actually used, rather all USB signals are
29 carried by printed circuit board traces and non-standard connectors are used.
30 Control software can be utilized, such as in one configuration where only e.g. 480
31 Mbit/sec devices are used, and split transactions on the hub devices need not be

1 supported. Likewise, because the USB bus 11 is an internal bus in this version
 2 of the invention, it is known when IDE devices 12 can be inserted or removed.
 3 For example, devices 12 may only be inserted (plugged in) or removed
 4 (unplugged) when specified criteria are met (e.g., when the front panel 25 is
 5 removed from the data storage system 24). As such, there is no need to
 6 constantly poll the bus 11 to determine device insertion or removal when it is
 7 known that e.g., the front cover 25 is in place (in one example, use of a "hall
 8 effect" switch allows determination of when the front panel 12 has been
 9 removed). This reduces a substantial amount of bus overhead, which translates
 10 into higher throughput. In another version, based on standard USB
 11 implementations, routine polling for device insertion/removal takes place.

12
 13 The USB system herein allows easy connection of a wide variety of
 14 devices (e.g., storage devices), providing user-friendly "hot plugging". The
 15 present invention allows "hot swap" of devices, whereby e.g. disk drives 12 can
 16 be plugged into the storage array 24 or removed without needing to power the
 17 system down or doing anything before the change is made.

18
 19 Hubs 16 may be used to associate up to 127 USB devices to a single USB
 20 controller 18 (multiple devices can easily be added to a single USB controller by
 21 chaining the devices together using hubs). As such, a hub is a USB device
 22 which extends the number of USB ports to connect other USB devices. The
 23 maximum number of attachable devices is reduced by the number of hubs on the
 24 bus. Hubs are self- and/or bus powered full speed devices. Typically the
 25 physical ports of the host controller are handled by a virtual root hub. This hub is
 26 simulated by the host controller's device driver and helps to unify the bus
 27 topology. As such, every port can be handled in the same way by the USB
 28 subsystem's hub driver.

29
 30 In one example, a USB controller 18 includes the root that is the signal
 31 source, and the storage devices have a downstream hub to terminate the signal.

1 The hubs 16 do all of the communication. Intermediate hubs can be included in
2 a configuration which can optionally provide additional power to their downstream
3 hubs or simply distribute what is provided by the upstream hub.
4

5 When a disk drive 12 with IDE to USB connector 18 is connected to the
6 USB controller/bus, the root hub senses the presence of the new device and
7 initially communicates with the device on e.g. "pipe 0," the default physical device
8 communications channel. Pipes are the data sub-bands of the hub architecture
9 that maintain the physical connections of devices. Once a device is recognized,
10 the root hub interrogates the device to find out what it is and what it is capable of
11 on pipe 0.
12

13 All of the devices 12 on the USB to IDE ports/connectors 18 are then
14 enumerated and each is assigned a unique device number, which also includes a
15 corresponding pipe number for physical device communications. The host 20
16 loads the program instructions to control the devices 12 and handles its
17 information flow. The hubs 16 are then running and information is passed in and
18 out of the USB controller 18 via the signal leads. The enumeration process is
19 initiated every time a device is plugged into or removed from the USB bus.
20

21 There are different types of USB devices as they can be used for different
22 purposes. First a device can be self powered, bus powered or both. The USB
23 can provide a power supply up to e.g. 500mA for its devices. If there are only
24 bus powered devices on the bus the maximum power dissipation could be
25 exceeded and therefore self powered devices exist. Such devices need to have
26 their own power supply. Devices that support both power types can switch to self
27 powered mode when attaching an external power supply. In examples where the
28 system is self-contained (e.g., FIG. 2), there is no need for "self powered bus"
29 feature of USB.
30

The maximum communication speed can differ for particular USB devices. The USB specification decides between low speed and full speed devices. Low speed devices (such as mice, keyboards, joysticks etc.) communicate at 1.5MBit/s and have only limited capabilities. Full speed devices (such as audio and video systems) can use up to 90% of the 12Mbit/s which is about 10Mbit/s including the protocol overhead. Further, USB 2.0 provides a transmission rate of 480 Mbits/sec, or a theoretical maximum of 60 Mbytes/sec. Disk devices can use the bulk message protocol of USB, and accounting for the protocol overhead, the theoretical maximum throughput can be 53 Mbytes/sec.

Using USB as an internal interconnect allows the use of inexpensive IDE disk drives within a drive array with simplified cabling/signal routing, "hot plugging", and totally passive center plane.

The present invention is applicable to different systems that utilize IDE devices (e.g., disk drives). In FIG. 4, the example data storage system 24 comprises three main components: a controller (e.g., host or CPU) 20, left rail disk drives 26 and right rail disk drives 28, in a housing 30. As shown in FIG. 3, each rail includes e.g. five disk drive packs 32, and each disk drive pack 32 includes e.g. three disk drives 12, wherein each disk drive 12 is an example of an IDE device. Referring to the example block diagram in FIG. 4, each IDE disk drive 12 in a drive pack 32 is to a USB to ATA protocol converter/bridge (i.e., IDE to USB controller) in the drive housing/carrier. The USB to ATA converters for each drive pack are connected to a USB hub, and the USB hubs are connected to USB host controllers as shown.

In this example, the data storage system 14 is in effect a "virtual library" system utilizing inexpensive ATA disk drives 12 to emulate a tape library. It is highly desirable to provide data storage (e.g., for data backup) as cheaply as possible, and to that end the use of ATA disk drives 12 is more attractive than either SCSI or Fibre channel disk drives.

1
2 Conventionally, USB architecture has been used for external attachment
3 of peripheral devices using physical cables. However, according to the present
4 invention, the USB architecture is utilized in the internal interconnect topology for
5 multiple IDE/ATA devices¹² (e.g., a disk array data storage device), alleviating
6 disadvantages of conventional ATA disk arrays. As such, the present invention
7 eliminates ATA cable/signal length limitations, provides hot plugging (lacking in
8 ATA specifications) and eliminates ATA signal routing issues. Using hot
9 plugging, individual disk drive carriers²² including a disk drive¹² and the
10 connected USB-to-IDE bridge¹⁴ (and optional hub¹⁶) can be
11 connected/disconnected to/from the data storage system²⁴. As shown in FIG.
12 1B, an HDD carrier²² can comprise a single USB to IDE bridge¹⁴ and its
13 associated disk drive¹², or the combination of a hub¹⁶ and several (e.g.,
14 2,3,...) USB to IDE bridges¹⁴ and their associated IDE disk drives¹².

15
16 USB supports hot plugging, and plug and play. Plug and play is
17 accomplished via device descriptors included in the USB standard which allow a
18 host system to query the type of device attached as well as model and vendor
19 specifics. This allows a host operating system to determine what devices are
20 attached and to load the proper device drivers automatically. Hot plugging is the
21 ability to attach (plug) or remove (unplug) devices from the system while the
22 system is running (operating). This is very important for disk arrays, because it
23 allows the replacement of failed components while the system is still running (i.e.,
24 without disturbing the normal operation of the host system). Conventional disk
25 array data storage systems using RAID protection can only tolerate one disk
26 drive failure within a RAID set, so the system is vulnerable to data loss for the
27 period of time after the first disk drive fails until it is replaced. Therefore it is
28 highly desirable to replace failed components as quickly as possible. Hot
29 plugging according to the present invention allows this replacement while the
30 system is still operational without waiting for the host to become quiescent.

FIGs. 5A-B and 6A-B show example flowcharts of embodiments of the method of the present invention. Flowcharts of FIGs. 5A-B are described herein in conjunction with FIG. 1A, wherein in one embodiment, the method the present invention includes the steps of: Providing multiple IDE devices 12 (step 40), providing multiple USB-to-IDE bridges 14 (step 42), connecting each IDE device 12 to a corresponding USB-to-IDE bridge 14 (step 44), providing at least one USB controller 18 (step 46) and connecting the bridges to the USB controller 18 (step 48). According to the steps in FIG. 5B, the system 10 operation is started (step 50), and thereafter an additional IDE device 12 is coupled to an additional USB-to-IDE bridge 14 (step 52) and connected to the system 10 (e.g., connect bridge 14 to the USB controller 18 or hub 16), while the system 19 is operation (step 54). Further, while the system 10 is operating an IDE device 12 is disconnected from the system 10 (e.g., disconnect IDE device 12 and corresponding USB-to-IDE bridge 14 from USB controller 18 or hub 16) (step 56).

Flowcharts of FIGs. 6A-B are described herein in conjunction with FIG. 1B, wherein in another embodiment, the method the present invention includes the steps of: Providing multiple IDE devices 12 (step 60), providing multiple USB-to-IDE bridges 14 and at least one hub 16 (step 62), connecting each IDE device 12 to a corresponding USB-to-IDE bridge 14 and connecting two or more bridges 14 to each hub 16 (step 64), providing at least one USB controller 18 (step 66) and connecting each hub 16 to the USB controller 18 (step 68). According to the steps in FIG. 6B, the system 10 operation is started (step 70), and thereafter additional IDE devices 12 coupled to corresponding USB-to-IDE bridges 14 and hub 16 (step 72) are connected to the system 19 (e.g., connect hub 16 to the USB controller 18), while the system is operation (step 74). Further, while the system 10 is operating one or more IDE devices 12 are disconnected from the system 10 (e.g., disconnect combination of IDE devices 12 corresponding USB-to-IDE bridges 14 and hub 16, from USB controller 18) (step 76).

1
2
3
4
5
6

The present invention has been described in considerable detail with reference to certain preferred versions thereof; however, other versions are possible. Therefore, the spirit and scope of the appended claims should not be limited to the description of the preferred versions contained herein.